อาจหาญ สัตยารักษ์

# ค.ศ. 2000 อันตรายจริงหรือ

## *เสนอแนวคิดในการแก้ไขปัญหาซอฟต์แวร์ที่เกี่ยวกับ เดือน-วัน-ปี สำหรับท่านที่เป็นโปรแกรมเมอร์*

ใคร ๆ ก็กล่าวขวัญถึงปี ค.ศ. 2000 ผู้เขียนซึ่งอยู่ในวงการคอมพิวเตอร์เช่นเดียวกันก็อยากจะให้ความเห็นของตนเองบ้าง แต่...เผอิญงานอิสระที่ผู้เขียนทำอยู่ไม่ยอมให้มีเวลาว่างที่จะเขียนถึงเลย วันนี้จึงขอเขียนถึงบ้าง ผู้อ่านลองมาพิจารณาความเห็นของผู้เขียนดูบ้าง จะเชื่อหรือไม่เชื่อก็ทดสอบได้ด้วยตนเอง

ความเห็นของผู้เขียนอาจจะไม่เหมือนความเห็นผู้อื่นอยู่บ้าง

สำหรับปี ค.ศ. 2000 ที่มีปัญหากล่าวขวัญกันนั้น ผู้เขียนค่อนข้างเชื่อว่าผู้ปล่อยข่าวเรื่องนี้น่าจะเป็นซอฟต์แวร์เฮ้าส์มากกว่า เพราะถ้ายูสเซอร์ตกใจมากเท่าใด ค่าปรับปรุงหรือแก้ไขซอฟต์แวร์น่าจะเรียกร้องได้มากตามไปด้วย

อย่าเพิ่งเชื่อผู้เขียนในตอนนี้นะ ขอให้ฟังความเห็นที่ผู้เขียนเสนอและคิดตามไปด้วยกันดีกว่า

## ทดสอบผ่านดอส

เรามาทดสอบเรื่องวันเดือนปีในระบบปฏิบัติการดอสกันนิดหนึ่งก่อน ว่าขอบเขตที่เครื่องคอมพิวเตอร์ยุคปัจจุบันกำหนดไว้นั้น อยู่ในช่วงใด

คำสั่งที่เราจะใช้คือ date ซึ่งในรูปแบบที่ดอสกำหนดไว้นั้น ทำให้คนหลายคนเข้าใจผิดมาตลอดว่า ในระบบดอสนั้นยอมให้ป้อนปี ค.ศ. ได้แค่ 2 หลักเท่านั้น ทั้งนี้เพราะเมื่อใช้คำสั่ง date จะแสดงรูปแบบการป้อนเป็น mm-dd-yy ไม่ได้แสดงเป็น mm-dd-yyyy

ตัวอย่างที่ผู้เขียนลองป้อนผ่านดอสมีดังนี้

```
C:\>date
Current date is Sat 03-29-1997
Enter new date (mm-dd-yy): 12-31-79

Invalid date
Enter new date (mm-dd-yy): 12-31-2099
C:\>date
Current date is Thu 12-31-2099
Enter new date (mm-dd-yy): 1-1-2100

Invalid date
Enter new date (mm-dd-yy): 3-29-97
```

ผลของการใช้คำสั่ง date สองครั้งข้างต้น ทำให้ทราบว่าวันที่ 31 เดือนธันวาคม ค.ศ. 2099 เป็นวันพฤหัสบดี (Thursday) และเมื่อป้อนวันเดือนปีเป็น 1 มกราคม 2100 ปรากฏว่าดอสไม่ยอมรับการป้อนครั้งนี้ ซึ่งแสดงว่าในระบบปฏิบัติการดอสนั้น ยอมรับวันเดือนปีได้สูงสุดคือ 31 ธันวาคม 2099 คืออีก 103 ปี

ปี ค.ศ. 2099 คือ ปี พ.ศ. 2099 + 543 = 2642

สังเกตคำสั่งแรกผู้เขียนป้อน 1-1-79 ก็ไม่ได้เช่นกัน เพราะว่าวันที่ต่ำสุดคือ 1-1-80 แต่...ป้อนเป็น 1-1-2079 ได้

สรุปคือ วันเดือนปีสูงสุดที่คอมพิวเตอร์ยอมได้นั้น นานจนคนเขียนโปรแกรมและคนใช้โปรแกรมในปัจจุบันต่างล้มหายตายจากไปก่อนอย่างแน่นอน

ผู้อ่านคงเข้าใจนะว่า รูปแบบที่กำหนดเป็น mm-dd-yy นั้นสามารถป้อนเป็น mm-dd-yyyy นั่นคือ ในการป้อนรูปแบบ เดือนวันปี ควรฝึกป้อนให้ครบ 10 หลักกันเสียตั้งแต่ตอนนี้

## ทดสอบผ่านปาสคาล

ลองมาตรวจสอบผ่านโปรแกรมภาษากันบ้าง ผู้เขียนขอยกตัวอย่างภาษาปาสคาลสักหนึ่งโปรแกรม ซึ่งหนังสือ "รวมข้อมูล PC สำหรับโปรแกรมเมอร์"

โดย ทอม โฮแกน บริษัท ซีเอ็ดยูเคชั่น จำกัด เป็นผู้พิมพ์และจำหน่าย โดยผู้เขียนเรียนขออนุญาตทางผู้แต่งและบริษัทฯ มาพร้อมนี้ด้วย ที่ขออ้างถึงบางส่วนในหนังสือ เพื่อที่ผู้อ่านจะได้หามาศึกษาเพิ่มเติมได้เอง (ราคาจำหน่ายตามปกหนังสือ 590 บาท)

มาดูรายละเอียดในหนังสือหน้า 2-25 หัวข้อ 2.20 DATW/TIME FORMATS จะแจ้งให้ทราบดังนี้

- ใช้ฟังก์ชัน 2AH, 28H คือกำหนดค่าเป็น $2A, $28 (ตัวอักษร H แสดงให้ทราบว่ากำหนดเป็นเลขฐานสิบหก)
- เรียกใช้อินเตอร์รัพต์หมายเลข 21H (ในภาษาปาสคาลใช้คำสั่ง Msdos ได้เลย)
- ผลลัพธ์ที่ได้มาคือ

- รีจีสเตอร์ AL เก็บเลขแสดงชื่อวัน (0-Sunday ไปจนถึง 6-Saturday)
- รีจีสเตอร์ DL เก็บเลขแสดงวัน (1-31)
- รีจีสเตอร์ DH เก็บเลขแสดงเดือน (1-12)
- รีจีสเตอร์ CX เก็บเลขแสดงปี ค.ศ. (1980-2099)

นำความรู้ที่ได้มานี้ ลองเขียนโปรแกรมสั้น ๆ ดูบ้าง จะได้ดังโปรแกรมในรูปที่ 1

ผู้เขียนไม่ขออธิบายการทำงานของโปรแกรมนี้ แต่ขอให้ท่านดูผลที่ได้บนจอภาพเมื่อรันโปรแกรมดูเอง จะพบว่า วันเดือนปีที่อ่านได้จะแสดงตามปกติที่เรารู้จักกันดี คือ

```
Today is Saturday   =   3/29/1997
                        mm-dd-yyyy
```

นั่นคือสามารถตรวจสอบปี พ.ศ. ได้ครบ 4 หลัก ดังนั้น...จึงสามารถตรวจสอบได้ถึง ปี ค.ศ. 2099 เช่นกัน

จากสองตัวอย่างที่ผ่านมา แสดงว่าปี ค.ศ. 2000 จะไม่ทำอันตรายใด ๆ กับเครื่องคอมพิวเตอร์ของเราอย่างแน่นอน เราผู้เป็นเจ้าของเครื่องจะไม่มีโอกาสได้อยู่ดูผลกระทบของระบบวันเดือนปีของดอส เพราะว่า

```
PROGRAM TESTNO1;
uses crt,dos;
var ch:char;
PROCEDURE ShowDate;
const Dayname:array[0..6] of string=
      ('Sunday','Monday','Tuesday','Wednesday',
       'Thursday','Friday','Saturday');
var Regs:registers;
begin
  Regs.AH:=$2A;     Msdos(Regs);
  write('Today is ',DayName[Regs.AL],' = ');
  writeln(Regs.DH:2,'-',Regs.DL:2,'-',Regs.CX);
  gotoxy(21,wherey); writeln('mm-dd-YYYY');
end;
begin
  clrscr;
  ShowDate;
  ch:=readkey;
end.
```

**รูปที่ 1 โปรแกรมตรวจสอบวันเดือนปี**

เราตายจากไปก่อน ที่กล่าวมานี้ผู้เขียนหมายถึงเครื่องคอมพิวเตอร์เท่านั้น ยังไม่ได้พูดถึงผลกระทบต่อการใช้ซอฟต์แวร์

ดังนั้น ปี ค.ศ. 2100 ต่างหากที่เป็นปีวิกฤตของคอมพิวเตอร์ (ยุคนี้)

## ปัญหาที่ซอฟต์แวร์

ปัญหาเรื่องเครื่องคอมพิวเตอร์นั้นเราตัดทิ้งไปได้ ด้วยเหตุผลที่ผู้เขียนยกมาแล้วข้างต้น สำหรับปัญหาที่น่ามองที่สุดและเป็นเหตุผลที่ผู้เขียนเชื่อว่าผู้ปล่อยข่าวเรื่องปี ค.ศ. 2000 นั้นเป็นซอฟต์แวร์เฮ้าส์ คือ ปัญหาเรื่องซอฟต์แวร์ ซึ่งน่าสนใจมากที่สุด มาลองวิเคราะห์กันดู

ปัญหาด้านซอฟต์แวร์นั้นเกิดจากเหตุผลสำคัญ คือ

1. เราถูกความเชื่อเรื่องประหยัดหน่วยความจำในการเก็บข้อมูลครอบงำนานหลายปี

2. รูปแบบของโปรแกรมสำเร็จบางอย่างกำหนดไว้อย่างนั้น เช่น โปรแกรมฐานข้อมูลทั่ว ๆ ไป กำหนดให้เดือนวันปีเป็น 8 ไบต์ คือ mm-dd-yy

ตรงนี้ขอแทรกเรื่อง เดือน-วัน-ปี ในโปรแกรมฐานข้อมูลยอดฮิตนั้น เป็นเรื่องของการนำมาเรียงลำดับเพื่อหาว่าใครเกิดก่อนกัน (ใครแก่กว่ากันนั่นแหละ) เป็นหลัก ถ้ากำหนดเป็น วัน-เดือน-ปี จะเรียงลำดับได้ไม่ถูกต้อง (โปรดสังเกตผู้เขียนใช้คำว่าไม่ถูกต้องนะครับ ไม่ใช่ไม่ได้)

3. เราเป็นคนไทย เมื่อใช้รูปแบบ mm-dd-yy เรายังสามารถใช้ได้ถึง ปี พ.ศ. 2599 คือยังเหลือเวลาอีก 2599-2540 = 59 ปี เพราะคงกำหนดฐานข้อมูลเป็น 03-29-40 ซึ่งจะหมายถึง 29 มี.ค. 2540

ทั้งโปรแกรมเมอร์และยูสเซอร์ ก็คงต้องจากโลกไปก่อนปี พ.ศ. 2540

ถ้าฐานข้อมูลได้กำหนดรูปแบบ เดือน-วัน-ปี เป็นไทย ๆ คือปี พ.ศ. อยู่แล้ว การแก้ปัญหาในช่วงนี้ยังไม่มีความจำเป็นใด ๆ ทั้งสิ้น เพราะจะยังคงสามารถตรวจสอบวันเดือนปีได้ตรงความเป็นจริงอีกนาน

หากอยากเปลี่ยนแปลงจาก 8 ไบต์ ให้เป็น 10 ไบต์ จะเกิดอะไรบ้างนั้น มันขึ้นกับขนาดของข้อมูลเป็นสำคัญ มาลองดูฐานข้อมูลของประเทศกันบ้าง ประชากรทั้งประเทศมี 60 ล้านคน ตามข่าวใหญ่ในช่วงปีที่ผ่านมา ซึ่งในเนื้อข่าวทำให้เราทราบคนไทยกลุ่มหนึ่งนับจำนวนเลขผิดด้วย เพราะข่าวแจ้งว่าคนไทยคนที่ 60 ล้านนั้นมีเกือบ 10 คนทีเดียว (คือนับตำแหน่ง 60 ล้านซ้ำไม่น้อยกว่าสิบครั้ง)

ประชากร 1 คน เปลี่ยนเดือน-วัน-ปี จาก 8 ไบต์ มาเป็น 10 ไบต์ จึงต้องใช้หน่วยความจำมากขึ้นอีก 2 ไบต์ต่อคน เพิ่มหน่วยความจำทั้งสิ้น 60x2 ล้านไบต์ หากนับวันที่ย้ายเข้า/ย้ายออก ก็จะเพิ่มมากขึ้นไปอีก

แต่ถ้าในฐานข้อมูลได้กำหนดเป็นสากล คือใช้ ปี ค.ศ. ตรานี้ ผู้เขียนคงตอบว่า มีปัญหาแน่นอน ซึ่งหนีไม่พ้นที่ต้องแก้ปัญหา แต่... ไม่ยากเกินกว่าความสามารถของโปรแกรมเมอร์แน่นอน

## มาแก้ปัญหากันเถอะ

ในการแก้ปัญหา เดือน-วัน-ปี นี้ ผู้เขียนจะไม่พูดถึงเรื่องเปลี่ยนจาก 8 ไบต์ ให้เป็น 10 ไบต์ เพราะคงแก้ไขได้ไม่ยาก เพียงแต่เขียนโปรแกรมนำไปรันผ่านระบบเพื่อค้นหาเปลี่ยนแปลง โครงสร้างข้อมูลปรับปรุงเดือน-วัน-ปีให้ตรงความเป็นจริงในขนาด 10 ไบต์ ซึ่งคงต้องรันอย่างไม่หยุดพักกันหลายวัน โดยคงต้องให้ซอฟต์แวร์ชุดใหม่กับการอบรมให้ทราบถึงส่วนที่เปลี่ยนแปลงไป

ผู้เขียนจะกล่าวถึงเฉพาะในเรื่องที่การดำเนินงานที่ไม่เปลี่ยนโครงสร้างข้อมูลเลย กล่าวคือไม่ต้องเขียนซอฟต์แวร์ในการปรับปรุงข้อมูล ไม่มีการแนะนำการใช้ซอฟต์แวร์ใหม่ ซึ่งหมายถึงว่า ยูสเซอร์ยังคงทำงานไปตามปกติ เพียงแต่ซอฟต์แวร์เฮ้าส์นำโปรแกรมใหม่ (หน้าตายังเหมือนเดิมทุกประการ) มาลงให้เท่านั้น

ก่อนอื่นข้อตกลงในเรื่องความคิดให้ตรงกันก่อนดังนี้

1. รูปแบบยังคงเป็น เดือน-วัน-ปี (mm-dd-yy)

2. เดือน-วัน-ปี ตามข้อ 1 เป็นสากล คือ ปี ค.ศ. ซึ่งจะมีขอบเขตเพียง 1980-1999 เท่านั้น

3. ไม่เพิ่มขนาดของโครงสร้างข้อมูล แต่...อาจเปลี่ยนชนิดของข้อมูลเท่านั้น

4. ถ้าเป็นฐานข้อมูลที่เป็นของโปรแกรมฐานข้อมูลทั่ว ๆ ไป ไม่ครอบคลุมการป้อนตรง ๆ ในฟิลด์นั้น ๆ แต่เป็นการป้อนผ่านโปรแกรมที่เราสร้างให้เท่านั้น

## แนวทางการแก้ปัญหา

ผู้เขียนคงไม่แสดงการเขียนโปรแกรมให้ทั้งหมด แต่จะกล่าวถึงหลักการในการแก้ปัญหาเท่านั้น โดยขอนำตัวอย่างของภาษาปาสคาลมาให้เห็นภาพการป้อนและการอ่านข้อมูลที่เกี่ยวกับ เดือน-วัน-ปี ซึ่งในตัวอย่างนี้ ผู้เขียนไม่ได้แสดงการอ่านหรือเขียนลงฐานข้อมูลจริง ๆ แต่ขอแสดงเพียงการอ่าน จะกำหนดโดยตรงเท่านั้น เพื่อให้เห็นแนวการทำงานและวิธีแก้ไขปัญหา

หลักการจะมีดังนี้

1. ข้อมูล เดือน-วัน-ปี กำหนดเป็นข้อความขนาด 8 ไบต์ (ในภาษาปาสคาลนั้น ข้อความทุกข้อความจะมีมิติที่ 0 เป็นความยาวข้อความ)

2. ส่วนกั้น เดือน-วัน-ปี กำหนดเป็นเครื่องหมาย / (เพื่อให้เหมือนในโครงสร้างของฐานข้อมูลทั่ว ๆ ไป)

3. กำหนดเงื่อนไขการทำงานดังนี้

3.1 การเก็บข้อมูลเป็นแบบเดิม แต่ให้ป้อนครบ 10 ไบต์ แต่ตัดให้เหลือเพียง 8 ไบต์ โดยใช้หลักการดังนี้

> ปี ค.ศ. 1980-1999 ตัดเหลือ 80-99
> ปี ค.ศ. 2000-2079 ตัดเหลือ 00-79
> ปี ค.ศ. 1900-1979 ไม่ยอมให้ป้อน
> ปี ค.ศ. 2080-2099 ไม่ยอมให้ป้อน

ซึ่งโดยแนวคิดนี้จะสามารถใช้โปรแกรมได้ถึงวันที่ 12 ธันวาคม ค.ศ. 2079 คืออีก 80 ปีข้างหน้า

3.2 การอ่านข้อมูลที่เก็บไว้ก็จะใช้แนวคิดย้อนทาง คืออ่านข้อมูล 8 ไบต์ แล้วนำมาขยายให้เป็นข้อมูล 10 ไบต์ โดยเป็นดังนี้

> ข้อมูล ปี ค.ศ. เป็น 80-99 แสดงเป็น 1980-1999
> ข้อมูล ปี ค.ศ. เป็น 00-79 แสดงเป็น 2000-2079

ลองมาดูโปรแกรมตามแนวคิดข้างต้น โดยเป็นการดำเนินการผ่านจอภาพเท่านั้น ไม่ได้อ่านหรือเขียนจากฐานข้อมูลจริง ซึ่งหากเป็นฐานข้อมูลจริง ก็สามารถปรับปรุงได้ (ดูรูปที่ 2)

จากตัวอย่างโปรแกรมข้างต้นใช้หลักการง่าย ๆ คือ ทำการตัดต่อเปลี่ยนแปลงข้อมูล เดือน-วัน-ปี ให้เป็นไปตามความต้องการ โดยยังคงโครงสร้างเดิมไว้ทุกประการ

```
PROGRAM TESTNO2;
 uses crt;
 var ch:char;
     date:string[10];
 FUNCTION Tran_Date(st:string):string;
 var  Datest:string[8];
 begin
   Datest:=copy(st,1,6);
   Datest:=Datest+copy(st,9,2);
   Tran_Date:=Datest;
 end;
 PROCEDURE savedate;
 begin
  repeat
   repeat
      write('Enter date (mm/dd/yyyy) ');
      readln(date);
      if date='' then exit;
   until date[0]=#10;
   writeln('Date ',Date,'  :New date ',Tran_date(date));
  until ch=#255;
 end;
 FUNCTION TranDate(st:string):string;
 var  Datest:string[10]; no:byte; ck:integer;
 begin
   Datest:=copy(st,1,6);
   val(copy(st,7,2),no,ck);
   if no<80 then Datest:=Datest+'20'+copy(st,7,2)
            else Datest:=Datest+'19'+copy(st,7,2);
   TranDate:=Datest;
 end;
 PROCEDURE Readdate;
 begin
  date:='30/29/97';
  writeln('Date ',Date,'  :New date ',Trandate(date));
  date:='30/29/99';
  writeln('Date ',Date,'  :New date ',Trandate(date));
  date:='30/29/00';
  writeln('Date ',Date,'  :New date ',Trandate(date));
  date:='30/29/79';
  writeln('Date ',Date,'  :New date ',Trandate(date));
  date:='30/29/80';
  writeln('Date ',Date,'  :New date ',Trandate(date));
 end;
 begin
  clrscr;
  savedate;
  Readdate;
 end.
```

รูปที่ 2 การแก้ปัญหาในเรื่อง เดือน-วัน-ปี

เพื่อความเข้าใจที่ตรงกัน ผู้เขียนขออธิบายเพียง 2 ฟังก์ชัน คือ trandate, tran_date ดังนี้

1. ฟังก์ชัน tran_date ทำการเปลี่ยนข้อมูล 10 ไบต์ ให้เป็น 8 ไบต์

ข้อมูล 12/31/1999 จะเหลือ 12/31/99
ข้อมูล 12/31/2079 จะเหลือ 12/31/79

ฟังก์ชันนี้จึงใช้ในการเก็บข้อมูลลงฐานข้อมูล 8 ไบต์เดิม

2. ฟังก์ชัน trandate ทำการเปลี่ยนข้อมูล 8 ไบต์ ให้เป็น 10 ไบต์

ข้อมูล 12/31/99 จะเปลี่ยนเป็น 12/31/1999
ข้อมูล 12/31/79 จะเปลี่ยนเป็น 12/31/2079

จึงใช้ในการอ่านข้อมูลที่เก็บตามข้อ 1 มาแสดงใหม่ให้เป็น 10 ไบต์

โปรแกรมข้างต้นนี้ในช่วงการป้อน ผู้เขียนยังไม่ได้ตรวจสอบขอบเขตของวันเดือนปีให้ถูกต้อง หากนำไปใช้งานจริงต้องเพิ่มเติมเอง

โดยแนวคิดที่ผู้เขียนเสนอนี้ ไม่ยุ่งเกี่ยวกับโครงสร้างฐานข้อมูลเลย ทั้งไม่ต้องมีการแก้ไขข้อมูลเดิมอีกด้วย ยูสเซอร์ก็อาจไม่ต้องอบรมใหม่ และจะเห็นว่าจริง ๆ แล้ว ปี ค.ศ. 2000 นั้น ยังไม่มีอันตรายใด ๆ อย่างที่คิดกันมากนัก

สำหรับแนวคิดนี้สามารถช่วยท่านได้ถึงปี ค.ศ. 2079 เท่านั้น หลังจากนั้นต้องหาทางแก้ไขปัญหาเอง ห้ามเขียนจดหมายไปถามผู้เขียนนะครับ เพราะวันนั้นผู้เขียนคงไม่มีโอกาสตอบท่านแล้ว

## แก้ให้ถึงปี ค.ศ. 9899

สำหรับโปรแกรมเมอร์ที่มีความห่วงใยลูกหลานมาก ก็ต้องใช้วิธีนี้ ซึ่งจะสามารถใช้ได้นานจนถึงวันที่ 12 ธันวาคม ค.ศ. 9899 ทีเดียว (โดยข้อมูลเก่าไม่ต้องแก้ไขใด ๆ ทั้งสิ้น)

หลักการมีดังนี้

1. ถ้าเป็นโครงสร้างของโปรแกรมฐานข้อมูลสำเร็จ ต้องเปลี่ยนชนิดข้อมูลจาก Date เป็นข้อมูลชนิด Charector ก่อน การเปลี่ยนแปลงโครงสร้างนี้ ยังคงใช้ข้อมูลเดิมได้ เพราะจะได้ 03/29/97 ในฐานข้อมูลอย่างเดิม (เหตุที่ต้องเปลี่ยนชนิดข้อมูลก็เพื่อจะสามารถป้อนตรง ๆ ได้ - แต่ต้องเข้าใจวิธีป้อนด้วย)

2. อาศัยรูปแบบที่กำหนดข้อมูลเป็นหลักในการแก้ไข เพราะเราใช้ / เป็นตัวกั้น เดือน-วัน-ปี ซึ่งค่ารหัสแอสกีของ / คือ 47 (ค่า 47-28=19) ซึ่งจะเท่ากับ 2 หลักแรกของ ปี ค.ศ. พอดี

ถ้าเราเปลี่ยน / (เฉพาะไบต์ที่ 3) ไปเป็นอย่างอื่น เราจะได้ปี ค.ศ. ที่เพิ่มขึ้นด้วย เช่น

: รหัสแอสกี 59 (59-28=31) เราเก็บ 12:31/99 จะเท่ากับ 12/31/3199
\ รหัสแอสกี 92 (92-28=64) เราเก็บ 12\31/99 จะเท่ากับ 12/31/6499
~ รหัสแอสกี 126 (126-28=98) เราเก็บ 12~31/99 จะเท่ากับ 12/31/9899

```
PROGRAM TESTNO2;
uses crt;
var ch:char;
   date:string[10];
FUNCTION TranDate(st:string):string;
var  Datest:string[10]; no:word; stt:string;
begin
   Datest:=copy(st,1,6);
   no:=ord(st[3])-28;  str(no,stt);
   Datest:=Datest+stt+copy(st,7,2);
   Datest[3]:='/';
   Trandate:=Datest;
end;
PROCEDURE Readdate;
begin
   date:='12/31/99';
   writeln('Date ',Date,' :New date ',Trandate(date));
   date:='12031/99';
   writeln('Date ',Date,' :New date ',Trandate(date));
   date:='12:31/99';
   writeln('Date ',Date,' :New date ',Trandate(date));
   date:='121/99';
   writeln('Date ',Date,' :New date ',Trandate(date));
   date:='12~31/99';
   writeln('Date ',Date,' :New date ',Trandate(date));
end;
begin
   clrscr;
   Readdate;
   ch:=readkey;
end.
```

**รูปที่ 3 การอ่านข้อมูล เดือน-วัน-ปี ที่เก็บได้ถึง ปี ค.ศ. 9899**

3. สรุปคือ

ข้อมูลของปี ค.ศ. 1900-1999 คงรูปแบบเดิม เช่น 12/31/99 คือ 12/31/1999
ข้อมูลของปี ค.ศ. 3100-3999 เปลี่ยนรูปแบบ คือ 12:31/99 คือ 12/31/3199
ข้อมูลของปี ค.ศ. 6400-6499 เปลี่ยนรูปแบบ คือ 12\31/99 คือ 12/31/6499
ข้อมูลของปี ค.ศ. 9800-9899 เปลี่ยนรูปแบบ คือ 12~31/99 คือ 12/31/9899

ผู้เขียนขอเสนอเฉพาะวิธีอ่านข้อมูลรูปแบบใหม่นี้ ซึ่งจะได้โปรแกรมดังรูปที่ 3

สำหรับรูปแบบนี้ ถ้าไปดูกันตรง ๆ ที่ข้อมูลที่เก็บอาจทำให้งงได้พอสมควร แต่...เนื่องจากเราบังคับให้ยูสเซอร์อ่านและป้อนข้อมูลผ่านโปรแกรมอยู่แล้ว ดังนั้นยูสเซอร์คงไม่รู้ว่าลักษณะของข้อมูลประหลาดอย่างไร เพราะการอ่านการเก็บสามารถให้ผลลัพธ์ทั้งบนจอภาพหรือเครื่องพิมพ์ถูกต้องตามความเป็นจริง

แม้ว่าผู้เขียนเพียงเสนอแนวคิดเท่านั้น ก็หวังว่าผู้อ่านที่เป็นโปรแกรมเมอร์คงจะเห็นแนวทางแก้ปัญหานี้ได้บ้าง และคงสามารถปรับให้เข้ากับงานที่ตนเองรับผิดชอบได้

ส่วนท่านที่ได้แก้ไขโครงสร้างจาก 8 ไบต์ ไปเป็น 10 ไบต์แล้ว ก็ไม่ต้องกังวลกับแนวคิดที่ผู้เขียนเสนอนี้ ขอเพียงให้จดจำไว้เป็นบทเรียนว่า "ปัญหาที่เกิดขึ้นนั้น ถ้าเราใช้เวลาคิดไม่ตื่นตกใจ เราย่อมแก้ปัญหาได้ตรงความต้องการอย่างมีประสิทธิภาพ"

สำหรับฉบับนี้ต้องบอกว่าลาก่อน พบกันใหม่ในฉบับหน้าครับ....